Amendments to the Specification:

On page 4, please amend paragraph beginning on line 3 as follows:

Yet another benefit of the present invention is that the invention provides a relatively low cost and low latency solution [[to]] for implementing large Patricia trees in data structure. Prior to the invention large Patricia-tree data structures were not acceptable in certain designs due to long latency associated with searching large Patricia tree data structures.

On page 9, please amend paragraph beginning on line 19 as follows:

Figure 4 shows a graphical representation of a data structure stored in memories of the ~~protocol~~ Network processor 10. The data structure includes direct table (DT) having a plurality of entries, only one is shown as an area bracketed by horizontal lines and is connected to a Patricia tree structure having a plurality of nodes termed Pattern Search Control Block (PSCB). Each of the PSCB, containing routing decision information, is connected to one or more leaves, five of which are shown in Figure 4. Included in the leaves are decisions or actions to be taken relative to a packet which has been correlated against information stored in the tree structure. As stated previously, the correlation would be carried out by a picoprocessor (coprocessor) executing a pico instruction or pico code. In addition, the correlation may be carried out by other means without deviating from the teachings of the present invention.

On page 11, please amend paragraph beginning on line 12 as follows:

It should be noted that other entities in the packet could have been used to access the direct table and walk the tree associated with the entry. By way of example the entries in the ~~data~~ direct table would correlate with section A of the frame. The coprocessor exercising the pico code correlates the ~~data~~ direct table with the 16 bits of the particular packet. If an entry in the ~~data~~ direct table matches the entry in the ~~table~~ packet, the remaining 32 bits are used to walk the tree as follows: significant bits in the remaining 32 bits are matched against PCB1. If the bit is logical 0 then the action stored with 0 in the PSCB is performed. If the bit is logical 1, the action stored with 1 in the PSCB is performed. In essence, the information in the PSCB dictates whether the tree is walked along the path labelled "X" to PSCB 3 or along the path labelled "Y" to PSCB 2. The PSCB 2 would have information which points to the information in leaf 2 or leaf 1. If the "X" path is walked, the next node would be PSCB 3 then PSCB 4. In this example the threshold is set at 3 and since all the leaves are located under the threshold, the booster CAM would not be used.